Lecture 12 Indirect Observations 12-1

$\hat{l} = G(x)$

$F(l,x) = \hat{l} - G(x) = 0$

$F(l,x) \approx F(l^0,x^0) + \frac{\partial F}{\partial l}\Delta l + \frac{\partial F}{\partial x}\Delta x$

$l^0 + \Delta l = l + v$, $\Delta l = \underline{(l - l^0)} + v$

$\{\Delta l_i = l_i^0 - l_{i-1}^0\}$ for I/O l^0 terms disappear

$F(l,x) \approx F(l^0,x^0) + \frac{\partial F}{\partial l}[(l-l^0)+v] + \frac{\partial F}{\partial x}\Delta x$

$= l^0 - G(x^0) + (I)[(l-l^0)+v] + B\Delta = 0$

$= l^0 - G(x^0) + l - l^0 + v + B\Delta = 0$

$= \underbrace{l - G(x^0)}_{F(l,x^0)} + v + B\Delta = 0$

Right Side Term is Misclosure should be small

$v + B\Delta = -F(l,x^0)$

$v + B\Delta = f$

Sep 21-4:26 PM

$V + B\Delta = f$ n obs, n_0 min ($u = n_0$), r redundancy 12-2

$W = \sigma_0^2 \Sigma^{-1}$

need B, f, W

$\Delta = (B^T W B)^{-1} B^T W f$

$X^0_{new} = X^0_{old} + \Delta$

examine magnitude of Δ if small, quit
if not small, re-linearize + iterate

2 approaches to check convergence

1. look at magnitude of each element of Δ
2. look for stability of $V^T W V$

after convergence:

$V = f - B\Delta$

$\hat{l} = l + v$

Sep 21-4:27 PM

for each iteration, make sure to evaluate 12-3

B with x^0

f with $-F(\underline{\ell}, \underline{x}^0)$

Observations Only $\frac{n}{\frac{n_0}{r}}$

$c = r$ nonlinear condition eqn's.

$F(\hat{\ell}) = 0$, $\underline{\underline{c = r}}$

$$F(\hat{\ell}) \approx F(\ell^0) + \frac{\partial F}{\partial \ell}\Delta\ell = 0$$

$$\Delta\ell = (\ell - \ell^0) + v$$

$$F(\hat{\ell}) = F(\ell^0) + A\left[(\ell - \ell^0) + v\right] = 0$$

$$A v = -F(\ell^0) - A(\ell - \ell^0)$$

$$A v = f$$

ℓ^0: usually much better quality than x^0

Sep 21-4:27 PM

$\underset{(c,n)}{A}\ \underset{(n,1)}{v} = \underset{(c,1)}{f}$, W, Q : inverses 12-4

$$Q_e = AQA^T$$

$$k = W_e f \text{ , } Q_e, W_e \text{ inverses}$$

$$v = QA^T k$$

$$\ell^0_{new} = \ell + v$$

reevaluate A, f + next iteration

$\Delta\ell = \ell^0_{new} - \ell^0_{previous}$

$\uparrow$ test for convergence ...

Sep 21-4:27 PM

Convergence check

12-5

Sample matlab code for handling iterations

all(v) : true if all elements of v are true

1 = true, 0 = false

```
niter = 0
keep_going = 1
while (keep_going == 1)
    LS code B, f, w, δ, update x
    if all( abs(del) < threshold)        del = δ
        keep_going = 0
        disp('we have converged'),
        v = f - Bδ,
        ...
        end
    if (niter > 10)
        keep_going = 0
        disp(' we did not converge')
        end
    niter = niter + 1
    end
```

Sep 21-4:27 PM